IN THE CLAIMS

Please amend the claims as follows.

1. (Previously Presented) A computer system comprising:

a network;

one or more processing nodes connected via the network, wherein each processing node includes:

a plurality of processors, wherein each processor includes a scalar processing unit, a vector processing unit, means for operating the scalar processing unit independently of the vector processing unit, a processor cache and a translation look-aside buffer (TLB), wherein the scalar processing unit places instructions for the vector processing unit in a queue for execution by the vector processing unit and the scalar processing unit continues to execute additional instructions; and

a shared memory connected to each of the processors within the processing node, wherein the shared memory includes a Remote Address Translation Table (RTT), wherein the RTT has capacity to store all physical page numbers associated with the processing node and wherein the RTT translates memory addresses received from other processing nodes such that the memory addresses are translated into physical addresses within the shared memory;

wherein processors on one node can load data directly from and store data directly to shared memory on another processing node via addresses that are translated on the other processing node using the other processing node's RTT; and

wherein each TLB translates memory references from its associated processor to the shared memory within the processing node.

2. (Canceled)

3. (Original) The computer system of claim 1, wherein the shared memory further includes a plurality of cache coherence directories, wherein each processing node is coupled to one of the cache coherence directories.

4. (Original) The computer system of claim 1, wherein each processor includes two vector pipelines.

5. (Original) The computer system of claim 1, wherein the processing nodes include at least one input/out (I/O) channel controller, wherein each I/O channel controller is coupled to the shared memory of the processing node.

6. (Previously Presented) The computer system of claim 1, wherein each scalar processing unit contains a scalar cache memory, wherein the scalar cache memory contains a subset of cache lines stored in the processor cache associated with its respective processor.

7. (Original) The computer system according to claim 1, wherein the network includes a router connecting one or more of the processing nodes.

8. (Previously Presented) A computer system comprising:

a network;

one or more processing nodes connected via the network, wherein each processing node includes:

four processors configured as a Multi-Streaming Processor, wherein each processor includes a scalar processing unit, a vector processing unit, means for operating the scalar processing unit independently of the vector processing unit, a processor cache connected to each of the processing units and a translation look-aside buffer (TLB), wherein the scalar processing unit places instructions for the vector processing unit in a queue for execution by the vector processing unit and the scalar processing unit continues to execute additional instructions; and

a shared memory connected to each of the processors within the processing node, wherein the shared memory includes a Remote Address Translation Table (RTT), wherein the RTT has capacity to store all physical page numbers associated with the processing node and wherein the RTT translates memory addresses received from other processing nodes such that the memory addresses are translated into physical addresses within the shared memory;

wherein processors on one node can load data directly from and store data directly to shared memory on another processing node via addresses that are translated on the other processing node using the other processing node's RTT; and

wherein each TLB translates memory references from its associated processor to the shared memory within the processing node.

9. (Canceled)

10. (Canceled)

11. (Original) The computer system of claim 8, wherein the shared memory further includes a plurality of cache coherence directories, wherein each processing node is coupled to one of the cache coherence directories.

12. (Previously Presented) A method of providing latency tolerant distributed shared memory multiprocessor computer system, wherein the method of providing comprising:

connecting one or more processing nodes via a network, wherein each processing node includes:

a plurality of processors, wherein each processor includes a scalar processing unit, a vector processing unit, means for operating the scalar processing unit independently of the vector processing unit, a processor cache and a translation look-aside buffer (TLB), wherein the scalar processing unit places instructions for the vector processing unit in a queue for execution by the vector processing unit and the scalar processing unit continues to execute additional instructions; and

a shared memory connected to each of the processors within the processing node, wherein the shared memory includes a Remote Address Translation Table (RTT), wherein the RTT has capacity to store all physical page numbers associated with the processing node;

storing data from a processor on a first processing node to shared memory on a second processing node via the network, wherein storing includes translating via the RTT on the second processing node memory addresses received from the first processing node such that the memory addresses received from the first processing node are translated into physical addresses within the shared memory of the second processing node; and

reading data from shared memory on the second processing node to the processor on the first processing node;

wherein memory references from the processor to the shared memory within the first processing node is translated by the associated TLB in the first processing node.

13. (Previously Presented) The method of claim 12, wherein each shared memory includes a plurality of cache coherence directories and wherein connecting includes coupling each processing node to one of the cache coherence directories.

14. (Previously Presented) The method of claim 12, wherein each processing node includes at least one input/out (I/O) channel controller and wherein connecting includes coupling each I/O channel controller to the shared memory of the processing node.

15. (Previously Presented) The method of claim 12, wherein each scalar processing unit includes a scalar cache memory and wherein connecting includes having the scalar cache memory contain a subset of cache lines stored in the processor cache associated with its respective processor.

16. (Previously Presented) The method of claim 12, wherein connecting includes routing one or more of the processing nodes through a router.

17. (Previously Presented) A method of providing latency tolerant distributed shared memory multiprocessor computer system, wherein the method of providing comprising:

connecting one or more processing nodes via a network, wherein each processing node includes:

four processors configured as a Multi-Streaming Processor, wherein each processor includes a scalar processing unit, a vector processing unit, means for operating the scalar processing unit independently of the vector processing unit, a processor cache connected to each of the processing units and a translation look-aside buffer (TLB), wherein the scalar processing unit places instructions for the vector processing unit in a queue for execution by the vector processing unit and the scalar processing unit continues to execute additional instructions; and

a shared memory connected to each of the processors within the processing node, wherein the shared memory includes a Remote Address Translation Table (RTT), wherein the RTT has capacity to store all physical page numbers associated with the processing node;

storing data from a processor on a first processing node to shared memory on a second processing node via the network, wherein storing includes translating via the RTT on the second processing node memory addresses received from the first processing node such that the memory addresses received from the first processing node are translated into physical addresses within the shared memory of the second processing node; and

reading data from shared memory on the second processing node to a the processor on the first processing node;

wherein memory references from the processor to the shared memory within the first processing node is translated by the associated TLB in the first processing node.

18. (Previously Presented) The method of claim 18, wherein the shared memory includes a plurality of cache coherence directories and wherein connecting includes coupling each processing node to one of the cache coherence directories.